Counting hexagonal patches and independent sets in circle graphs

Paul Bonsma[1]* and Felix Breuer[2]

[1] Technische Universität Berlin, Institut für Mathematik,
Sekr. MA 5-1, Straße des 17. Juni 136, 10623 Berlin, Germany.
bonsma@math.tu-berlin.de
[2] Freie Universität Berlin, Institut für Mathematik,
Arnimallee 3, 14195 Berlin, Germany.
felix.breuer@fu-berlin.de

Abstract. A hexagonal patch is a plane graph in which inner faces have length 6, inner vertices have degree 3, and boundary vertices have degree 2 or 3. We consider the following counting problem: given a sequence of twos and threes, how many hexagonal patches exist with this degree sequence along the outer face? This problem is motivated by the enumeration of benzenoid hydrocarbons and fullerenes in computational chemistry. We give the first polynomial time algorithm for this problem. We show that it can be reduced to counting maximum independent sets in circle graphs, and give a simple and fast algorithm for this problem.

Keywords: graph algorithms, computational complexity, counting problem, planar graph, circle graph, fullerene, hexagonal patch, fusene, polyhex.

1 Introduction

The notions used and problems introduced in this section are defined more formally in Section 2. A plane graph G is a graph together with a fixed planar embedding in the plane. The unbounded face is called the *outer face* and the other faces are called *inner faces*. The boundary of the outer face is simply called the *boundary* of G. A *hexagonal patch* is a 2-connected plane graph in which all inner faces have length 6, boundary vertices have degree 2 or 3, and non-boundary vertices have degree 3. These graphs are also known as *fusenes* [18], *hexagonal systems* [10], *polyhexes* [17] and $(6,3)$-*polycycles* [12] in the literature. These graphs are well-studied in mathematical and computational chemistry since they model benzenoid hydrocarbons and graphite fragments (see e.g. [18] and the references therein). A central question is that of enumerating hexagonal patches, either of a given size [6], or with a given boundary form.

A sequence $x_0,\dots,x_{k-1}$ of twos and threes is a *boundary code* of a hexagonal patch G if there is a way to label the boundary vertices of G with $v_0,\dots,v_{k-1}$ such that $v_0,\dots,v_{k-1},v_0$ is a boundary cycle of G, and the degree $d(v_i)=x_i$ for all i. Note

* Both authors are supported by the Graduate School “Methods for Discrete Structures” in Berlin, DFG grant GRK 1408.

that cyclic permutations and/or inversions of the sequence can yield different boundary codes for the same patch, but for the question we study this fact is not important. It is well-known and easily observed using Euler's formula that the boundary code of a hexagonal patch satifies $d_2 - d_3 = 6$, where d_i is the number of boundary vertices with degree i. We define the parameters $d_2(X)$ and $d_3(X)$ also for sequences X of twos and threes, as expected. One may ask whether a hexagonal patch exists that satisfies a given boundary code. A result by Guo, Hansen and Zheng [18] shows that this question is not as easy as was first expected: in Figure 1 their example is shown which shows that different patches may exist with the same boundary code. This can be verified by comparing the degree of v_1 with u_1, v_2 with u_2, etc. Our drawing of this graph is taken from [8]. (In [18] it is also shown that although multiple solutions may exist, they all have the same size.) Therefore the following question should be asked: given a sequence S of twos and threes, *how many* hexagonal patches exist with boundary code S? This counting problem is called *Hexagonal Patch*. Guo et al [18] and Graver [17] give conditions for when solutions are unique, if they exist. Deza et al [10] give an algorithm for deciding whether at least one solution exist. The complexity of their algorithm is however superexponential. In addition they give a polynomial time algorithm for a very restricted case (see Section 3). These results have been generalized to various problem variants, mainly by varying the conditions on the face lengths and vertex degrees, see e.g. [12, 7–9]. However, the question whether the counting problem can be answered efficiently remained open.

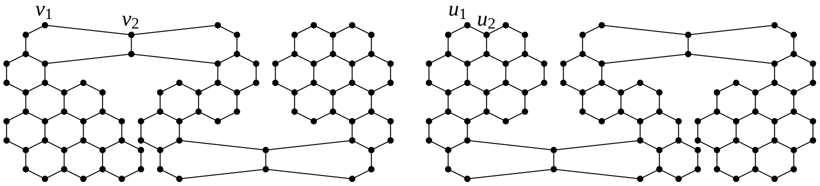


Fig. 1. Two different patches with the same boundary code.

In this paper we show that the counting problem Hexagonal Patch can be solved in time $O(k^3)$ where k is the length of the sequence. This is surprising since the number of solutions may be exponential in k, as can easily be seen by generalizing the example from Figure 1. Therefore, we can only return the *number* of solutions in polynomial time, and not return a *list* of all corresponding patches. The algorithm can however be extended to generate all patches in time $k^{O(1)} \cdot O(n)$, where n is the number of returned solutions. We remark that it is not hard to generalize our result to the generalizations introduced in [8]: A 2-connected plane graph is an (m,k)-*patch* if all inner faces have length k, inner vertices have degree m and boundary vertices have degree at most m. Our methods work for instance for $(4,4)$ and $(6,3)$-patches in addition to $(3,6)$-patches, but for simplicity we restrict to hexagonal patches.

An additional motivation for this result is the following: *fullerene patches* generalize hexagonal patches by also allowing 5-faces in addition to 6-faces. Such patches model fragments of fullerene molecules, and therefore their enumeration is another important problem from computational chemistry. Fullerene molecules have at most twelve 5-faces. The current result is an essential ingredient for the result we give in a second paper [2], where we give a polynomial time algorithm for deciding whether a given boundary code belongs to a fullerene patch with at most five 5-faces.

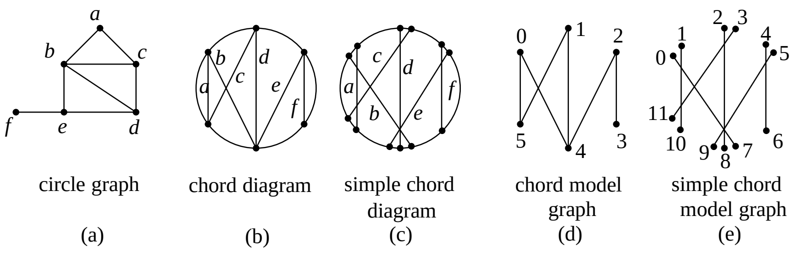
Our algorithm is based on the following idea: with a few intermediate steps, we transform the problem Hexagonal Patch to the problem of counting maximum independent sets in circle graphs. A *circle graph* G is the intersection graph of chords of a circle (detailed definitions are given below). Algorithms are known for the optimization problem of finding maximum independent sets in circle graphs [16], but counting problems on circle graphs have not been studied to our knowledge.

circle graph

In this paper we give a simple dynamic programming algorithm for counting independent sets in circle graphs. In addition this algorithm improves the complexity for the optimization problem. Circle graphs can be represented as follows (see Figure 2(a),(b)): Every vertex of G is associated with a *chord* of a circle drawn in the plane, which is a straight line segment between two points on the circle, such that two vertices are adjacent if and only if the two chords overlap (possibly only in a common end). We will represent chord diagrams with graphs as follows (see Figure 2(d)). Number the points on the circle that are ends of chords with $0,\dots,k-1$, in order around the circle, and view these as vertices. View a chord from i to j as an edge ij. Call the resulting graph G' the *chord model graph*. Note that (maximum) independent sets of the circle graph correspond bijectively to *(maximum) planar matchings* or (M)PMs of G', which are (maximum) matchings M that do not contain edges $\{i,j\}$ and $\{x,y\}$ with $i<x<j<y$. Hence counting MPMs in G' is polynomially equivalent to counting maximum independent sets in circle graphs.

chord model graph

(maximum) planar matchings



circle graph
(a)

chord diagram
(b)

simple chord diagram
(c)

chord model graph
(d)

simple chord model graph
(e)

Fig. 2. A circle graph, (simple) chord diagram and (simple) chord model graph.

Circle graphs are extensively studied and generalize permutation graphs and distance hereditary graphs, see e.g. [5]. Recognizing them and constructing a chord representation can be done in polynomial time [4, 15], and the current fastest algorithm uses time $O(n^2)$, where n is the number of vertices [22]. A number of problems that are NP-hard on general graphs are easy on circle graphs, such as in particular finding maximum independent sets [16, 23, 24, 19].

The first algorithm for the optimization problem by Gavril [16] has time complexity $O(m^3)$, where m is the number of vertices of the circle graph, which is the number of edges of the corresponding chord model graph. This was improved to $O(m^2)$ by Supowit [23]. Recently this has been improved further by Valiente [24] in the way we will explain now. All of these algorithms work with the chord model graph (or chord diagram), and as a first step, transform it into a 1-regular graph as shown in Figure 2(e): for a vertex of degree d, d new vertices are introduced, and the d incident edges are distributed among these in such a way only one of these edges can appear in a PM of G. This does not change the size and number of MPMs. The resulting graph G has $2m$ vertices and m edges, and is called the *simple chord model graph*. We assume the vertices are numbered $0,\dots,2m-1$, in the proper order. Then the *length* of an edge $ij \in E(G)$ is $|j-i|$. The algorithm from [24] has complexity $O(l)$, where l is the sum of all edge lengths of the simple chord model graph obtained this way. Clearly this is at most $O(m^2)$, and in many cases better. However, when *dense* chord model graphs are given on n vertices and $m \in \Omega(n^2)$ edges, this algorithm may need $\Omega(n^4)$ steps. Our transformation from Hexagonal Patch yields a chord model graph G', which in fact may be dense.

We give a simple algorithm with complexity $O(nm)$, which not only determines the size of a MPM, but also counts the number of MPMs of the chord model graph. This improvement in time complexity is possible by working with arbitrary degrees, and not using the simple chord model graph, in contrast to all previous algorithms for this problem [24].

The outline of the paper is as follows. In Section 2 we give definitions, and a precise formulation of the problem. In Section 3 we define locally injective homomorphisms to the hexagonal lattice (the brickwall) as a way of representing problem instances and solutions and reduce the counting problem to a problem on walks in the brickwall. In Section 4 we reduce that problem to that of counting *proper assignment sets* of the walk, which is in fact the problem of counting MPMs in chord model graphs. In Section 5 we present our algorithm for counting MPMs, and in Section 6 we give a summary of our algorithm for Hexagonal Patch. We end in Section 7 with a discussion, where we also discuss a similar problem from topology. Statements for which proofs are omitted are marked with a star, the proofs appear in the appendix.

2 Preliminaries

walk
internal
end
closed
path
cycle
k-cycle
W_x

For basic graph theoretic notions not defined here we refer to [11]. A *walk* of *length* k in a (simple) graph G is a sequence of $k+1$ vertices $v_0,\dots,v_k$ such that v_i and v_{i+1} are adjacent in G for all $i \in \{0,\dots,k-1\}$. $v_1,\dots,v_{k-1}$ are the *internal* vertices and v_0, v_k the *end* vertices of the walk. The walk is *closed* if $v_0 = v_k$. Throughout this paper we will in addition assume that $v_{i-1} \neq v_{i+1}$ for all $i \in \{1,\dots,k-1\}$, and if the walk is closed, $v_1 \neq v_{k-1}$ (i.e. we will assume walks *do not turn back*). If $v_i \neq v_j$ for all $i \neq j$ then the walk is a *path*. If the walk is closed and $v_i \neq v_j$ for all distinct $i,j \in \{0,\dots,k-1\}$ then it is also called a *cycle*. A cycle of length k is also called a *k-cycle*. For a walk $W = v_0,\dots,v_k$, W_x denotes v_x. If W is a closed walk, then W_x denotes $v_{x \bmod k}$. We will also talk about the *vertices* and *edges* of a walk, which are defined as expected. In a slight abuse of terminology, the graph consisting of these vertices and edges will also be called a walk (or path or cycle if applicable).

Let H be a hexagonal patch, and B be a boundary cycle of H of length k. Let $X = x_0, \dots, x_{k-1}$ be a sequence of twos and threes. We say that the tuple (H,B) is a *solution for the boundary code X* if $d(B_i) = x_i$ for all $i \in \{0, \dots, k-1\}$. Two solutions (H,B) and (H',B') are considered *equivalent* if there is an isomorphism ψ from H to H' such that $\psi(B_i) = B'_i$ for all i. Formally, when we ask for the number of *different* pairs (H,B) that satisfy some property, we want to know how many equivalence classes contain a pair (H,B) satisfying this property. The counting problem Hexagonal Patch is now defined as follows: given a sequence X, how many different solutions (H,B) to X exist?

equivalent

3 From Boundary Codes to Walks in the Brickwall

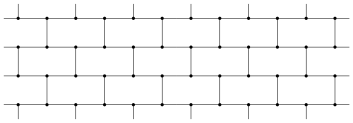


Fig. 3. The brickwall $\mathscr{B}$.

An (infinite) 3-regular plane graph where every face has length 6 is called a *brickwall*. It can be shown that the facial cycles are the only 6-cycles of a brickwall, and that all brickwalls are isomorphic.

brickwall

We will use $\mathscr{B}$ to denote the brickwall as drawn in Figure 3. Edges that are horizontal (vertical) in this drawing are called the *horizontal* (*vertical*) edges of $\mathscr{B}$. Paths consisting of horizontal edges are called *horizontal paths*. Two vertices joined by a horizontal path are said to have the same *height*.

$\mathscr{B}$
horizontal
vertical
horizontal paths
height

The reason that we study brickwalls is because the following mapping of hexagonal patches into them is very useful. Let H be a hexagonal patch. A *locally injective homomophism (LIH)* of H into $\mathscr{B}$ is a mapping of the vertices of H to vertices of $\mathscr{B}$, such that adjacent vertices are mapped to adjacent vertices in $\mathscr{B}$, and such that all neighbors of any vertex in H are mapped to different vertices in $\mathscr{B}$. Since the shortest cycles in $\mathscr{B}$ are of length 6, a LIH into $\mathscr{B}$ maps 6-cycles to 6-cycles. Since the faces of $\mathscr{B}$ are the only 6-cycles in $\mathscr{B}$, we see that a LIH of H into $\mathscr{B}$ also maps inner faces to faces.

locally injective homomophism (LIH)

Loosely speaking, the idea behind these mappings is as follows. Let H be a hexagonal patch of which we fix a boundary cycle B. When we map H with a LIH ϕ into $\mathscr{B}$, then the boundary B is mapped to some walk W in $\mathscr{B}$. But now it can be shown that this walk W is only determined by the choice of the initial vertices and the boundary code of H. Hence instead of asking how many hexagonal patches exist with a certain boundary code, we may ask how many patches exist that can be mapped properly to the brickwall, such that the boundary coincides with the walk that is deduced from the boundary code. Below we will go into more detail.

The technique of mapping patches to brickwalls is not new, and is actually considered folklore to some extent [10]. For instance, Deza et al [10] observe that Hexagonal Patch can be solved in polynomial time if the LIH is bijective, and Graver [17] shows that the problem Hexagonal Patch can only have multiple solutions if there is a brickwall vertex that has at least three preimages in such a LIH. We will however study these mappings more in more detail than has been done before, and develop new concepts, and prove new statements which we feel are of independent interest.

W makes a right (left) turn at i

Let W be a walk in a 3-regular plane graph G. We say W *makes a right (left) turn at i* when edge W_iW_{i-1} immediately follows edge W_iW_{i+1} in the clockwise (anticlockwise) order around W_i. Note that since we assume that walks do not turn back and G is 3-regular, W makes either a left or a right turn at every i.

Walk construction: Using a given sequence $x_0,\dots,x_{k-1}$ of twos and threes, we construct a walk $W = v_0,\dots,v_k$ in $\mathscr{B}$ as follows. For v_0 and v_1, choose two (arbitrary) adjacent vertices. For $i \geq 1$, choose v_{i+1} such that W makes a left turn at i if $x_i = 3$, and makes a right turn at i if $x_i = 2$.

solution for W
equivalent

Let W be a closed walk in $\mathscr{B}$ of length k, H be a hexagonal patch, ϕ a LIH from H to $\mathscr{B}$ and B a boundary walk of H of length k. Then the tuple (H,ϕ,B) is said to be a *solution for W* when $\phi(B_i) = W_i$ for all i. Two solutions $S = (H,\phi,B)$ and $S' = (H',\phi',B')$ are considered to be *equivalent* if and only if there is an isomorphism ψ from H to H' such that $\psi(B_i) = B'_i$ for all i. We say that ψ is an (or demonstrates the) equivalence between S and S'. The LIH ϕ allows us to use the terminology defined for $\mathscr{B}$ for the graph H as well; we will for instance call edges of H *horizontal* or *vertical* if their images under ϕ are horizontal or vertical, respectively.

clockwise solution
$\textsc{Right}(W)$
$\textsc{Left}(W)$
turning number
$t(W)$

Let the boundary B of a hexagonal patch H be mapped to the closed walk W in $\mathscr{B}$ by the LIH ϕ. This is a *clockwise solution* if and only if for every i, $d(B_i) = 2$ if W makes a right turn at i, and $d(B_i) = 3$ if W makes a left turn at i. It is *anticlockwise* when these conditions are reversed. Let $\textsc{Right}(W)$ and $\textsc{Left}(W)$ denote the number of indices $i \in \{0,\dots,k-1\}$ such that W makes a right turn or left turn at i, respectively. The *turning number* of W is $t(W)$= $(\textsc{Right}(W) - \textsc{Left}(W))/6$. Using the fact that for a solution (H,ϕ,B), ϕ maps faces of H to faces of $\mathscr{B}$, it can be shown that every solution is either clockwise or anticlockwise. Since a hexagonal patch has $d_2 - d_3 = 6$ (d_i is the number of degree i vertices on the boundary), Lemma 1 then follows. Variants of Lemma 2 have been proved in [8, 17].

Lemma 1 (*) *Let W be a closed walk in $\mathscr{B}$. If $t(W) = 1$, then every solution to W is clockwise. If $t(W) = -1$ then every solution to W is anticlockwise. If $t(W) \notin \{-1,1\}$, then no solution exists.*

Lemma 2 *Let (H,B) be a solution to a boundary code X and let W be a walk in $\mathscr{B}$ that is constructed using X. Then there exists a unique LIH ϕ such that (H,ϕ,B) is a clockwise solution to W.*

Because of Lemma 2, we may rephrase the problem Hexagonal Patch in terms of solutions (H,ϕ,B) to a closed walk W in the brickwall.

Theorem 3 *The number of different (hexagonal) solutions for a boundary code X with $d_2(X) - d_3(X) = 6$ is the same as the number of different clockwise solutions for the walk W in $\mathscr{B}$ that is constructed using X.*

Proof: For any solution (H,B) for X, a unique LIH ϕ exists such that (H,ϕ,B) is a clockwise solution to W (Lemma 2). For any clockwise solution (H,ϕ,B) to W, the characterization of clockwise solutions and the construction of W shows that (H,B) is a solution to X. (Since $d_2(X) - d_3(X) = 6$ and $t(W) = 1$ by Lemma 1, W turns at 0 as prescribed by x_0.) Note that the definitions of equivalence for pairs (H,B) and triples (H,ϕ,B) coincide and, in particular, do not depend on ϕ. □

4 From Walks in the Brickwall to Assignment Sets

Throughout Section 4, W denotes a closed walk in $\mathscr{B}$ with length k. We first sketch the main idea of this section. If we consider a solution (H,ϕ,B) to W, then we mentioned above that this defines which edges of H are horizontal and vertical. Now if we start at a boundary vertex B_i of H that is incident with a horizontal interior edge of H, then we can continue following this horizontal path of H until we end in a different boundary vertex B_j. We will say that this solution *assigns* i to j. If we only know all assignments defined by the solution this way, we can reconstruct the unique solution. We will deduce properties of such sets of assignments such that there is a solution if and only if these properties are satisfied. The purpose is to show that we may focus on counting such assignment sets instead of solutions to the walk.

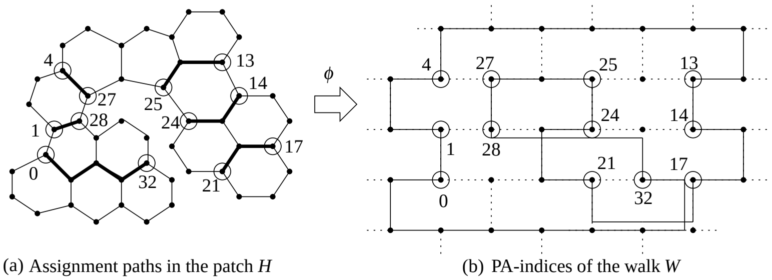
For all i, j where W_i and W_j lie on the same height, $H_{i,j}$ denotes the horizontal walk in $\mathscr{B}$ from W_i to W_j. Consider an index $i \in \{0,\ldots,k-1\}$ and the vertex W_i. Let u be the neighbor of W_i in $\mathscr{B}$ not equal to W_{i-1} or W_{i+1}. If u has the same height as W_i and W makes a left turn at i, then index i is called a *PA-index*. In Figure 4(b) an example is shown, where vertices corresponding to PA-indices are encircled, and their indices are shown. Note that if W has a clockwise solution (H,ϕ,B), then the PA-indices are precisely those indices i such that B_i has degree 3 and the interior edge incident with B_i is horizontal (see Figure 4(a)).

$H_{i,j}$

PA-index

A *possible assignment (PA)* is a pair $\{i, j\}$ of PA-indices with $W_i \neq W_j$ such that W_i and W_j have the same height and $H_{i,j}$ does not contain any of $W_{i-1}, W_{i+1}, W_{j-1}, W_{j+1}$ (note that $H_{i,j}$ has non-zero length). For instance, in Figure 4(b) some PAs are $\{1,28\}$, $\{1,14\}$ and $\{21,32\}$, but $\{1,24\}$ is not.

possible assignment (PA)



(a) Assignment paths in the patch H

(b) PA-indices of the walk W

Fig. 4. Assignment paths of a solution and PA-indices of a walk.

assignment path

Let (H,ϕ,B) be a clockwise solution to a closed walk W in $\mathscr{B}$. An *assignment path* P is a horizontal path in H from B_i to B_j where $i \neq j$, and all edges and internal vertices of P are interior edges and vertices of H. In Figure 4(a) the assignment paths of the given solution are shown in bold.

Proposition 4 (*) *If a clockwise solution (H,ϕ,B) to W contains an assignment path from B_i to B_j, then $\{i,j\}$ is a PA of W.*

assigns i to j

$\mathscr{A}(S)$

assignment set

This motivates the following definition. A clockwise solution $S=(H,\phi,B)$ to a walk W *assigns i to j* if there is an assignment path from B_i to B_j. For each clockwise solution S, we define the set $\mathscr{A}(S) := \{\{i,j\} : \{i,j\}$ is a PA of W and S assigns i to $j\}$. This is the *assignment set* defined by the solution S.

Lemma 5 (*) *Let W denote a closed walk in $\mathscr{B}$ and let S,S' be clockwise solutions of W. If S and S' are equivalent, then $\mathscr{A}(S) = \mathscr{A}(S')$.*

Now we will deduce the properties of a set $\mathscr{A}(S)$. Proposition 6 shows that assignment paths do not share vertices. Combining this with planarity yields Proposition 7.

Proposition 6 (*) *Let (H,ϕ,B) be a clockwise solution to W. Every interior vertex of H and every vertex B_i, where i is a PA-index, lies on a unique assignment path.*

Proposition 7 (*) *Let S be a solution to a closed walk W that assigns i to j. For any x,y with $x < i < y < j$ or $i < x < j < y$, S does not assign x to y.*

perfect matching

non-crossing

assignment set

proper assignment set

These two propositions give us properties a set of the form $\mathscr{A}(S)$ for a clockwise solution S necessarily has to have. Given W, a set A of possible assignments of W is a *perfect matching* on the set of PA-indices if for every PA-index i of W there is exactly one pair $\{i,j\} \in A$. A is *non-crossing* if there do not exist assignments $\{i,j\},\{x,y\} \in A$ such that $i < x < j < y$. An *assignment set* for W is a set of possible assignments of W. It is a *proper assignment set* if it is a non-crossing, perfect matching on the set of PA-indices of W. Combining Proposition 4, Proposition 6 and Proposition 7 yields Lemma 8. Lemma 9 states more or less the reverse; the long proof appears in Appendix B.

Lemma 8 *If $S=(H,\phi,B)$ is a clockwise solution of W then $\mathscr{A}(S)$ is a proper assignment set for W.*

Lemma 9 (*) *Let W denote a closed walk in $\mathscr{B}$ with $t(W)=1$, and let A be a proper assignment set of W. Then there exists a clockwise solution S of W with $\mathscr{A}(S)=A$.*

It remains to establish the converse of Lemma 5. Suppose we have two solutions $S=(H,\phi,B)$ and $S'=(H',\phi',B')$ with $\mathscr{A}(S)=\mathscr{A}(S')$. Every vertex of H and H' lies on the boundary or on an assignment path (Proposition 6). Therefore we can use the boundary and the assignment paths to define a bijection $\psi : V(H) \to V(H')$. When doing this appropriately, it can be shown that ψ an equivalence.

Lemma 10 (*) *Let W be a closed walk in $\mathscr{B}$, and let S and S' denote clockwise solutions of W. If $\mathscr{A}(S)=\mathscr{A}(S')$, then S and S' are equivalent.*

Theorem 11 *Let W be a walk in $\mathscr{B}$ with $t(W) = 1$. The number of equivalence classes of solutions to W is the same as the number of different proper assignment sets for W.*

Proof: The above lemmas show that $S \mapsto \mathscr{A}(S)$ gives a bijection from the set of equivalence classes of clockwise solutions of W to the set of proper assignment sets for W, since the following properties are satisfied: (1) *$\mathscr{A}$ is well-defined:* Let S_1 and S_2 denote clockwise solutions of W. If S_1 and S_2 are equivalent, then $\mathscr{A}(S_1) = \mathscr{A}(S_2)$ (Lemma 5). (2) *The range of $\mathscr{A}$ is correct:* For any clockwise solution S of W the set $\mathscr{A}(S)$ is a proper assignment set for W (Lemma 8). (3) *$\mathscr{A}$ is injective:* Let S_1 and S_2 denote clockwise solutions of W. If $\mathscr{A}(S_1) = \mathscr{A}(S_2)$, then S_1 and S_2 are equivalent (Lemma 10). (4) *$\mathscr{A}$ is surjective:* For any proper assignment set A for W, there exists a clockwise solution S of W with $\mathscr{A}(S) = A$ (Lemma 9). □

It follows that for solving the Hexagonal Patch problem, we may focus on counting proper assignment sets for the walk W (assuming $t(W) = 1$).

5 Counting Maximum Planar Matchings

In this section we will observe that the remaining algorithmic problem is that of counting independent sets in circle graphs, and present a fast algorithm for this problem. We use the closed walk W in $\mathscr{B}$ to construct a graph G^W with vertex set $V = \{0, \dots, n-1\}$, where n is the number of PA-indices of W. Let $p_0, \dots, p_{n-1}$ be all PA-indices of W, numbered according to their order in W. Then the edge set of G^W will be $E = \{ij \mid \{p_i, p_j\}$ is a PA of $W\}$. The following lemma is now easily observed.

Lemma 12 *Let G^W be the graph as constructed above from the walk W. If G^W has no perfect planar matching, then W has no proper assignment sets. Otherwise the number of proper assignment sets for W is equal to the number of MPMs in G^W.*

Now we will present an algorithm for counting MPMs of a graph G with $V(G) = \{0, \dots, n-1\}$. As mentioned in the introduction, this is equivalent to counting maximum independent sets in a circle graph H, where G is the chord model graph of H. We will present this algorithm for the general case where G has edge weights: w_{ij} denotes the edge weight of ij, and a PM M is *maximum* if $\sum_{e \in M} w_e$ is maximum.

For $i, j \in V(G)$ with $i \leq j$, let $G_{i,j} = G[\{i, \dots, j\}]$. If $i > j$, then $G_{i,j}$ is the empty graph. For $i, j \in V(G)$, let $S_{i,j}$ denote the size of a MPM in $G_{i,j}$. In particular, $S_{0,n-1}$ is the size of a MPM in G. We now give a subroutine $S(i, j)$ for calculating $S_{i,j}$, which considers the sizes of various PMs for $G_{i,j}$, and returns the size of the largest PM.

$G_{i,j}$

$S_{i,j}$

A subroutine $S(i, j)$ for calculating $S_{i,j}$:

```
$m := 0$
**If** $i < j$ **then** $m := S_{i+1,j}$
**For** $v \in N(i)$ with $i+1 \leq v \leq j-1$:
    $m := \max\{m, S_{i,v} + S_{v+1,j}\}$
**If** $j \in N(i)$ and $j > i$ **then**
    $m := \max\{m, w_{ij} + S_{i+1,j-1}\}$
Return $m$
```

Lemma 13 (*) *Let G be a graph with $V(G) = \{0,\dots,n-1\}$ and $i, j \in V(G)$. If the values $S_{x,y}$ are known for all x,y with $-1 \le y - x < j - i$, then the subroutine $S(i, j)$ computes $S_{i,j}$ in time $O(d(i))$.*

$N_{i,j}$ Let $N_{i,j}$ denote the number of MPMs in $G_{i,j}$. Below is a similar subroutine $N(i, j)$ for calculating $N_{i,j}$, which considers various PMs for $G_{i,j}$, checks whether they are maximum by comparing the size with $S_{i,j}$, and keeps track of the number of MPMs using the variable N.

A subroutine $N(i, j)$ for calculating $N_{i,j}$:

```
If $j \le i$ then Return 1, exit.
$N := 0$
If $S_{i,j} = S_{i+1,j}$ then $N := N + N_{i+1,j}$
For $v \in N(i)$ with $i+1 \le v \le j-1$:
    If $S_{i,j} = S_{i,v} + S_{v+1,j}$ then $N := N + N_{i,v} \times N_{v+1,j}$
If $j \in N(i)$ and $S_{i,j} = w_{ij} + S_{i+1,j-1}$ then $N := N + N_{i+1,j-1}$
Return $N$
```

Lemma 14 (*) *Let G be a graph with $V(G) = \{0,\dots,n-1\}$ and $i, j \in V(G)$. If the values $S_{x,y}$ and $N_{x,y}$ are known for all x,y with $-1 \le y - x < j - i$, and $S_{i,j}$ is known, then the subroutine $N(i, j)$ computes $N_{i,j}$ in time $O(d(i))$.*

Theorem 15 *Let G be a graph with $V(G) = \{0,\dots,n-1\}$ on m edges. The size and number of MPMs of G can be computed in time $O(nm)$.*

Proof: For $d = -1$ to $n-1$, we consider all $i, j \in \{0,\dots,n-1\}$ with $j - i = d$, and calculate $S_{i,j}$ and $N_{i,j}$ using the above subroutines. This way, for every value of d, every vertex of G is considered at most once in the role of i. For this choice of i, calculating $S_{i,j}$ and $N_{i,j}$ takes time $O(d(i))$ (Lemma 13, Lemma 14). Hence for one value of d this procedure takes time $O(\sum_{i \in V(G)} d(i)) = O(m)$. □

We remark that Valiente's algorithm [24] for simple (1-regular) chord model graphs can also be extended by using Subroutine $N(i, j)$ to calculate $N_{i,j}$ in constant time, immediately any time after a value $S_{i,j}$ is calculated. This then yields time complexity $O(l)$ and space complexity $O(n)$. In some cases it may be better to transform to a simple chord model graph and use this algorithm.

6 Summary of the Algorithm

We now summarize how counting the number of hexagonal patches that satisfy a given boundary code X of length k can be done in time $O(k^3)$. W.l.o.g. $d_2(X) - d_3(X) = 6$. First use X to construct a walk W in $\mathscr{B}$ of length k, as shown in Section 3. Theorem 3 shows that we may now focus on counting clockwise solutions to W. If W is not closed it clearly has no solution. Since $d_2(X) - d_3(X) = 6$ we may now assume $t(W) = 1$. Then Theorem 11 shows we may focus on counting proper assignment sets for W. Now construct G^W as shown in Section 5. G^W has n vertices where $n < k$ is the number of PA-indices of W (and $O(n^2)$ edges). By Lemma 12, the number of proper assignment sets for W is equal to the number of MPMs of G^W, provided that G^W has a perfect PM. This number and property can be determined in time $O(n^3) \in O(k^3)$ (Theorem 15).

7 Discussion

Our first question is whether the complexity of $O(k^3)$ can be improved. Secondly, considering the motivation from benzenoid hydrocarbons, it is interesting to study whether a patch exists that has a 'reasonably flat' embedding in $\mathbb{R}^3$ using *regular* hexagons. More precisely, this is the brickwall walk problem, but requires in addition giving a consistent linear order ('depth') for all vertices mapped to the same vertex of $\mathscr{B}$. It may also be interesting to study generalizations such as to surfaces of higher genus.

After we presented an early version of this work [3], Jack Graver pointed us to a similar well-studied problem in topology. Let S^1 denote the unit circle and D^2 the unit disk in $\mathbb{R}^2$. An *immersion* is a continuous function $f : A \to B$ such that for every x in A there is a neighborhood N of x such that $f|_N$ is a homeomorphism. (A *curve* when $A = S^1$, $B = \mathbb{R}^2$.) An immersion $c : S^1 \to \mathbb{R}^2$ of the circle into the plane is *normal* if c has only finitely many double-points and c crosses itself at each of these. Two immersions d, d' are *equivalent* if there exists a homeomorphism $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ such that $d \circ \phi = d'$. Now the *Immersion Extension* problem is this: given an immersion $c : S^1 \to \mathbb{R}^2$, how many immersions $d : D^2 \to \mathbb{R}^2$ exist that extend c? Note that this problem is not combinatorial, therefore it makes no sense to study its computational complexity. One can turn it into a combinatorial problem by restricting the input to *piecewise linear (PL)* curves $c : S^1 \to \mathbb{R}^2$.

When viewing the walk constructed in Section 3 as a curve, there are obvious similarities between the Hexagonal Patch problem and the Immersion Extension problem. However, to our knowledge it is an open problem to prove that these problems are in fact equivalent. The ideas introduced here may be helpful for giving such a proof. Establishing this would provide insight to both problems, since the Immersion Extension problem is well-studied – at least on normal curves – see e.g. [1, 13, 21]. Interestingly, Blank [1, 14] reduces the Immersion Extension problem problem to a combinatorial problem that is essentially the same as counting MPMs in simple chord model graphs. He does not address the complexity of this problem. Shor and Van Wyk [21] were the first to study the complexity of the combinatorial Immersion Extension problem on normal curves. They give an $O(n^3 \log n)$ algorithm where n is the number of pieces of the PL curve c. Assuming the equivalence of the Immersion Extension problem and the Hexagonal Patch problem, this would give an alternative algorithm for Hexagonal Patch; note that there are methods for transforming general PL curves to equivalent normal PL curves [20]. Since our algorithm does not need such a step, it is not only faster but also much easier to implement (see also [20]). However, the question of equivalence of these problems is still interesting because many generalizations of the Immersion Extension problem have been studied [13]. Finally, we believe that in fact our method can be adapted to give a simple and fast for the combinatorial Immersion Extension problem that does not require the assumption that the given curve is normal, but that is beyond the scope of this paper.

Acknowledgement We thank Gunnar Brinkmann for introducing us to this subject and his suggestions, and Hajo Broersma for the discussions on this topic.

References

1. S.J. Blank. *Extending immersions of the circle*. PhD thesis, Brandeis University, 1967.
2. P. Bonsma and F. Breuer. Finding fullerene patches in polynomial time. submitted.
3. P. Bonsma and F. Breuer. Finding fullerene patches in polynomial time I: Counting hexagonal patches. http://arxiv.org/abs/0808.3881v1, 2008.
4. A. Bouchet. Reducing prime graphs and recognizing circle graphs. *Combinatorica*, 7(3):243–254, 1987.
5. A. Brandstädt, V.B. Le, and J.P. Spinrad. *Graph classes, a survey*. SIAM, 1999.
6. G. Brinkmann, G. Caporossi, and P. Hansen. A survey and new results on computer enumeration of polyhex and fusene hydrocarbons. *J. Chem. Inf. Comput. Sci.*, 43:842–851, 2003.
7. G. Brinkmann and B. Coppens. An efficient algorithm for the generation of planar polycyclic hydrocarbons with a given boundary. *MATCH Commun. Math. Comput. Chem.*, 2009.
8. G. Brinkmann, O. Delgado-Friedrichs, and U. von Nathusius. Numbers of faces and boundary encodings of patches. In *Graphs and discovery*, volume 69 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 27–38. Amer. Math. Soc., Providence, RI, 2005.
9. G. Brinkmann, J.E. Graver, and C. Justus. Numbers of faces in disordered patches. *Journal of mathematical chemistry*, 45:263–278, 2009.
10. M. Deza, P. W. Fowler, and V. Grishukhin. Allowed boundary sequences for fused polycyclic patches and related algorithmic problems. *J. Chem. Inf. Comput. Sci.*, 41:300–308, 2001.
11. R. Diestel. *Graph theory*. Springer-Verlag, Berlin, third edition, 2005.
12. M. Dutour Sikirić, M. Deza, and M. Shtogrin. Filling of a given boundary by p-gons and related problems. *Discrete Applied Mathematics*, 156:1518–1535, 2008.
13. D. Eppstein and E. Mumford. Self-overlapping curves revisited. http://arxiv.org/abs/0806.1724, 2008.
14. G.K. Francis. Extensions to the disk of properly nested plane immersions of the circle. *Michigan Math. J.*, 17(4):377–383, 1970.
15. C.P. Gabor, K.J. Supowit, and W.L. Hsu. Recognizing circle graphs in polynomial time. *Journal of the ACM*, 36(3):435–473, 1989.
16. F. Gavril. Algorithms for a maximum clique and a maximum independent set of a circle graph. *networks*, 3(3):261–273, 1973.
17. J. E. Graver. The (m,k)-patch boundary code problem. *MATCH Commun. Math. Comput. Chem.*, (48):189–196, 2003.
18. X. Guo, P. Hansen, and M. Zheng. Boundary uniqueness of fusenes. *Discrete Applied Mathematics*, 118:209–222, 2002.
19. N. Nash, S. Lelait, and D. Gregg. Efficiently implementing maximum independent set algorithms on circle graphs. *Journal of experimental algorithmics*, 13, 2009.
20. R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete and Computational Geometry*, 19(1):1–17, 1998.
21. P.W. Shor and C.J. Van Wyk. detecting and decomposing self-overlapping curves. *Computational Geometry*, 2:31–50, 1992.
22. J.P. Spinrad. Recognition of circle graphs. *Journal of algorithms*, 16(2):264–282, 1994.
23. K.J. Supowit. Finding a maximum planar subset of a set of nets in a channel. *IEEE Transactions on computer-aided design*, 6(1):93–94, 1987.
24. G. Valiente. A new simple algorithm for the maximum-weight independent set problem on circle graphs. In *ISAAC 2003*, volume 2906 of *LNCS*, pages 129–137, 2003.

A Proofs of Section 3 and Section 4

Proof of Lemma 1: Let ϕ be a LIH from a hexagonal patch H with boundary cycle B of length k to $\mathscr{B}$, and let $W = \phi(B_0), \phi(B_1), \ldots, \phi(B_{k-1}), \phi(B_0)$. We first show that (H, ϕ, B) is either a clockwise solution to the walk W or an anticlockwise solution to W.

An index i is called *locally clockwise* if either $d(B_i) = 2$ and W makes a right turn at i, or $d(B_i) = 3$ and W makes a left turn at i.

We show that if some i is locally clockwise, then every index is locally clockwise. Suppose this is not true, so then there is an i that is locally clockwise such that $i+1$ is not. Assume first that $d(B_i) = 2$ and $d(B_{i+1}) = 2$. Then W makes a right turn at i, but a left turn at $i+1$. Therefore W_{i-1} and W_{i+2} do not lie at a common facial cycle of $\mathscr{B}$. Since B_i and B_{i+1} both have degree 2 in H, all of the vertices $B_{i-1}, \ldots, B_{i+2}$ lie at a common inner face of H. This is a contradiction since ϕ maps faces of H to faces of $\mathscr{B}$.

In the case where $d(B_i) = d(B_{i+1}) = 3$, we consider the neighbor v_i of B_i that is not equal to B_{i-1} or B_{i+1}, and the neighbor v_{i+1} of B_{i+1} that is not equal to B_i or B_{i+2}. These again lie at a common face of H, but if W makes a left turn at i and a right turn at $i+1$, are mapped to two vertices that do not lie at a common face, which again yields a contradiction. The two other cases are analogous. We conclude that if a solution contains a locally clockwise vertex, it is clockwise.

Now we relate this to the turning number. Let d_i denote the number of vertices of degree i on the boundary of H. If (H, ϕ, B) is a clockwise solution then $d_2 = \textsc{Right}(W)$ and $d_3 = \textsc{Left}(W)$. We know that $d_2 - d_3 = 6$ since H contains no 5-faces. Hence $t(W) = 1$. Similarly, if an anticlockwise solution exists then $t(W) = -1$ follows, which proves the statement. □

Proof of Proposition 4: We show that if a clockwise solution (H, ϕ, B) to W assigns B_i to B_j, then $\{i, j\}$ is a PA of W. Let P be an assignment path from B_i to B_j. We have that B_i and B_j have the same height, since P is horizontal. $B_i \neq B_j$ holds since H is 2-connected. P is then mapped to a non-zero length path in $\mathscr{B}$ (it does not turn back, since ϕ is a LIH), so $W_i \neq W_j$ follows. All edges of P are interior edges of H, so B_i and B_j have degree 3, and therefore W makes a left turn at i and j. It follows that i and j are PA-indices. Since ϕ is a LIH and P contains no boundary edges, $H_{i,j}$ does not contain any of $W_{i-1}, W_{i+1}, W_{j-1}, W_{j+1}$. □

Proof of Lemma 5: We show that if two clockwise solutions S and S' of W are equivalent, then $\mathscr{A}(S) = \mathscr{A}(S')$. Let $S = (H, \phi, B)$ and $S' = (H', \phi', B')$, and let $\psi : V(H) \to V(H')$ demonstrate their equivalence. Note that $\phi' \circ \psi$ and ϕ are both LIHs from H into $\mathscr{B}$ that map B_i to W_i. As the LIH with this property is uniquely determined by Lemma 2, we conclude that $\phi' \circ \psi = \phi$. In particular, any edge uv is horizontal in H if and only if $\psi(u)\psi(v)$ is horizontal H'. Clearly an analogous statement holds for vertices being interior. Therefore ψ maps assignment paths to assignment paths. Since $\psi(B_i) = B'_i$ for all i it follows that $\mathscr{A}(S) = \mathscr{A}(S')$. □

Proof of Proposition 6: Let (H, ϕ, B) be a clockwise solution to W. We show that every interior vertex of H and every vertex B_i, where i is a PA-index, lies on a unique assignment path. Let M be the set of horizontal non-boundary edges of H, and let $H' =$

$(V(H), M)$. Since ϕ is a LIH, H' has maximum degree at most 2. H' contains no cycles, because these would have to be mapped to cycles of $\mathscr{B}$ but $\mathscr{B}$ contains no cycles with only horizontal edges. Hence H' is a set of paths and isolated vertices.

It can be seen that vertices with degree 2 in H' are interior vertices of H, and that vertices with degree 1 in H' are equal to B_i for some PA-index i. Hence the path components of H' (paths of non-zero length) are all assignment paths. Since all assignment paths in H are also part of H', we see that there is a one-to-one correspondence between assignment paths in H and non-trivial components of H'. We also see that every interior vertex of H and every vertex B_i where i is a PA-index lies on one such path. The statement follows. □

Proof of Proposition 7: We show that if a solution (H, ϕ, B) assigns i to j and $i < x < j < y$, then it does not assign x to y. Suppose H contains an assignment path P from B_i to B_j, and an assignment path Q from B_x to B_y. By Proposition 6, P and Q have no vertices in common. But since the (distinct) end vertices of the paths appear in the order B_i, B_x, B_j and B_y along a boundary cycle of the plane graph H, this is impossible. (Formally, to obtain a contradiction, we may use P, Q and the boundary cycle of H to exhibit a subdivision of K_4 that is embedded with all vertices on the boundary, which then would yield a planar embedding of K_5.) □

horizontal left
horizontal right

For the proof below and later proofs in the appendix, it is important to distinguish between two different kinds of horizontal edges of $\mathscr{B}$: *horizontal left* (*horizontal right*) edges are edges that follow a vertical edge after turning left (right). Note that this partitions the edges of $\mathscr{B}$ into vertical edges, horizontal left edges, and horizontal right edges, and that every face contains two of each. The same holds for faces in a solution (H, ϕ, B); Recall that if patch H is mapped by a LIH ϕ to $\mathscr{B}$, this allows us to define vertical and horizontal (left / right) edges in H. Similarly, we will talk about vertices of H that lie to the left / below etc. other vertices. This is also defined by ϕ and the chosen drawing of $\mathscr{B}$.

Proof of Lemma 10: We show that if $\mathscr{A}(S) = \mathscr{A}(S')$ for two solutions S and S', then these solutions are equivalent. Let $S = (H, \phi, B)$ and $S' = (H', \phi', B')$. We construct the isomophism ψ from H to H' that will demonstrate the equivalence as follows. For all i, $\psi(B_i) = B'_i$. This defines ψ for boundary vertices. Every non-boundary vertex lies on a unique assignment path (Proposition 6). Suppose such a vertex v lies on an assignment path P from B_i to B_j. Then $\{i, j\} \in \mathscr{A}(S)$ and thus $\{i, j\} \in \mathscr{A}(S')$. The assignment path P' from B'_i to B'_j in H' is also mapped by ϕ' to $H_{i,j}$ and therefore has the same length as P (since ϕ and ϕ' are LIHs). Now if v is the x-th vertex on P, ψ will map v to the x-th vertex of P'. This defines ψ. Since every vertex of H lies on the boundary or on an assignment path, the function ψ is defined for every vertex of H, and since the same holds for H', ψ is a bijection. By definition ψ maps boundary vertices to the correct boundary vertices, so to demonstrate that ψ is an equivalence between S and S', it only remains to show that it is an isomorphism.

We only show that edges of H are mapped to edges of H' by ψ. By symmetry a similar statement then follows for ψ^{-1}, which proves that ψ is an isomorphism. Clearly ψ maps boundary edges of H to boundary edges of H'. Observe that every horizontal non-boundary edge of H lies on an assignment path. Therefore ψ also maps horizontal edges of H to edges of H'. What remains are vertical edges of H that do not lie on the boundary. Note that ψ maps horizontal left (right) edges of H to edges of H' of the same type, and by observing the same for ψ^{-1}, it also follows that if a vertical edge is mapped to an edge, it is mapped to a vertical edge again.

Suppose there exists a (vertical, interior) edge of H that is not mapped to an edge of H' by ψ. Let $e = u_1v_1 \in E(H)$ be such an edge such that all edges that lie to the left of it are mapped to edges of H'. Suppose u_1 lies below v_1. e is incident with two inner faces of H, so we may choose $F = u_1,u_2,u_3,v_3,v_2,v_1,u_1$ to be the inner face of H on the left side of e. Note that all other edges of F are mapped to edges of H'; four edges are horizontal, and the other vertical edge is mapped by our choice of e.

Let u'_1, u'_2, u'_3, v'_3, v'_2 and v'_1 respectively be the images under ψ of the vertices of F. The edges $v'_1v'_2$ and $v'_2v'_3$ lie on a common face of H' (since v'_2 has degree at most 3). We show that they lie on a common *inner* face of H'. If not, then both $v'_1v'_2$ and $v'_2v'_3$ are boundary edges. Then the corresponding edges v_1v_2 and v_2v_3 of H are boundary edges too, and thus these two edges share both an inner face and the outer face. Since H is 2-connected, it follows that $d(v_2) = 2$. Because both H and H' are clockwise solutions, $d(v'_2) = 2$. Hence the two edges in H' also share two faces, and thus one inner face.

Let $F' = v'_1,v'_2,v'_3,x,y,z,v'_1$ be the inner face of H' on which these two edges lie. Since ψ maps to edges of the same type, $v'_1v'_2$ and $v'_2v'_3$ are horizontal left and horizontal right edges respectively. Since ϕ' maps inner faces of H' to inner faces of $\mathscr{B}$, v'_3x is a vertical edge. v'_3 is incident with at most one vertical edge (ϕ' is a LIH), so we may conclude that $x = u'_3$ (here we use the fact that vertical edges are not mapped to horizontal edges, so $v'_3u'_3$ is vertical). Continuing this reasoning shows that $F' = v'_1,v'_2,v'_3,u'_3,u'_2,u'_1,v'_1$. Hence $u'_1v'_1 \in E(H')$, a contradiction with the choice of u_1v_1. We conclude that ψ is an isomorphism, which concludes the proof. □

B The proof of Lemma 9

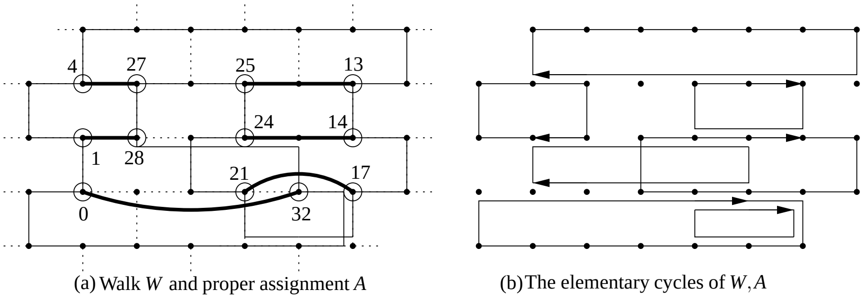
Before we can prove Lemma 9 we need to introduce some new terminology and lemmas. For a closed walk $W = v_0,\dots,v_k$, the *subwalk of W from i to j* is the walk $v_i,v_{i+1},\dots,v_j$ of length $j-i \bmod k$. If $j<i$ then, more precisely, this is the walk $v_i,v_{i+1},\dots,v_{k-1},v_0,\dots,v_j$. The subwalk of W from i to j will be denoted by $W_{i,j}$. If $W = v_0,\dots,v_k$ is a closed walk, then for any $i \in \{0,\dots,k-1\}$, the walk $W' = v_i,v_{i+1},\dots,v_{k-1},v_0,v_1,\dots,v_{i-1},v_i$ is called a *rotation* of W. We will write $W' \approx W$ to express that W' is a rotation of W. For a pair of walks $W = v_0,\dots,v_k$ and $W' = u_0,\dots,u_l$ with $v_k = u_0$ and $v_{k-1} \neq u_1$, $W \circ W'$ denotes the *concatenation* of W and W', which is $v_0,\dots,v_k,u_1,\dots,u_l$. In sequences, the notation $(a)^b$ means that b copies of a are inserted in the sequence at this point. For instance, $1,2,(3)^3,4$ denotes the sequence $1,2,3,3,3,4$.

$W_{i,j}$
rotation
$W' \approx W$
$W \circ W'$
concatenation
$(a)^b$

Let W be a closed walk in $\mathscr{B}$. For indices i, let $n(i)$ be the first PA-index after i (not equal to i). So with respect to the walk W shown in Figure 4(b), $n(1) = 4$, $n(32) = 0$, etc. The subwalks of the form $W_{i,n(i)}$ for any PA-index i are called the *pieces* of W.

$n(i)$
pieces

Elementary Cycles Let A be a proper assignment set of a closed walk W in $\mathscr{B}$. We will now define how such a tuple W,A gives *elementary cycles*, which correspond to closed walks in $\mathscr{B}$. Informally, for any index i the unique elementary cycle that contains W_iW_{i+1} can be found as follows. This is illustrated in Figure 5, where the proper assignment given by the solution shown in Figure 4(a) is used. Arcs are shown to indicate the direction of the elementary cycles, and the head of the arc indicates the first vertex.



(a) Walk W and proper assignment A

(b) The elementary cycles of W, A

Fig. 5. Elementary cycles in $\mathscr{B}$ given by a proper assignment A of W.

Start at W_i, and continue along W (in the direction of increasing indices) until the first PA-index $n(i)$ is found. Let $\{n(i), i_1\} \in A$ be the (unique) assignment in A that contains $n(i)$. Then continue along the horizontal path $H_{n(i),i_1}$ in $\mathscr{B}$. At W_{i_1}, follow W again until the next PA-index $n(i_1)$, and then follow $H_{n(i_1),i_2}$ where $\{n(i_1), i_2\} \in A$, etc. Continue with this procedure, taking alternatingly pieces of W and horizontal paths that correspond to assignments in A until we arrive again at W_i. Observe that this procedure ends and that we do actually arrive at W_i again this way. We consider the choice of starting vertex to be irrelevant, hence in the following formal definition we fix a canonical rotation. An *elementary cycle* of W, A is a walk in $\mathscr{B}$ of the form

elementary cycle

$$C = W_{i_0,n(i_0)} \circ H_{n(i_0),i_1} \circ W_{i_1,n(i_1)} \circ \ldots \circ H_{n(i_l),i_0},$$

where $\{n(i_j), i_{j+1}\} \in A$ for all j, and $\{n(i_l), i_0\} \in A$. In addition, we require that $i_0 < i_j$ for all $j \in \{1, \ldots, l\}$. This last condition fixes the canonical rotation. Note that elementary cycles that start at different indices of W may still yield the same walk in $\mathscr{B}$. Therefore we also consider a sequence I of numbers that give the corresponding indices of W. We insert -1 in this sequence for the vertices that correspond to vertices of horizontal paths instead of parts of W. For example, for the above choice of C,

$$I \;=\; i_0, \ldots, n(i_0) \;,\; (-1)^{x-1} \;,\; i_1, \ldots, n(i_1) \;,\; \ldots\ldots \;,\; (-1)^{y-1} \;,\; i_0,$$

where x and y are the length of $H_{n(i_0),i_1}$ and $H_{n(i_l),i_0}$ respectively. (Obviously, if $i_0 > n(i_0)$, then $i_0, \ldots, n(i_0)$ should be read as $i_0, i_0+1, \ldots, k-1, 0, 1, \ldots, n(i_0)$, where k is the length of W, etc.)

Formally, an elementary cycle is now a pair C, I of a closed walk C in $\mathscr{B}$ and sequence of numbers I that are of the form explained above. This formal definition is needed to clearly define what the *number* of elementary cycles of W, A is: elementary cycles are still considered different even if they yield the same walk, but admitting different rotations is irrelevant. However, below we will often informally denote elementary cycles just by C; the index sequence I is clear from how we denote C. Note that for every i there is a unique elementary cycle C, I such that I contains i and $i+1$ consecutively. In a slight abuse of notation, from now on we will often simply express this statement as follows: there is a unique elementary cycle C that contains the walk edge W_iW_{i+1}. In the special case where $A = \emptyset$, there is only one elementary cycle C, I, which has $C = W$.

A walk in $\mathscr{B}$ in which every vertical edge is followed by a horizontal right edge and preceded by a horizontal left edge is called a *right-turn walk*.

right-turn walk

Proposition 16 *Let A be a proper assignment set for a walk W in $\mathscr{B}$. Then every elementary cycle C of W, A is a right-turn walk.*

Proof: Any vertical edge of C must come from a piece of W. So let $W_{i-1}W_i$ be this vertical edge. If W turns left at i, then i is a PA-index and C turns right. If W turns right at i, then i is not a PA-index and C turns right as well. Hence in both cases, a vertical edge in C is followed by a horizontal right edge. Similarly, C turns right at $i-1$ in both the case that it is a PA-index and the case that it is not, hence vertical edges in C are preceded by horizontal left edges. □

For the following proof, we use the following vertex labelling for $\mathscr{B}$. See also Figure 6, which illustrates the next Proposition.

$$V(\mathscr{B}) = \{b_{i,j} : i, j \in \mathbb{Z}\}$$

$$E(\mathscr{B}) = \{b_{i,j}b_{i+1,j} : i, j \in \mathbb{Z}\} \cup \{b_{i,j}b_{i+1,j+1} : i, j \in \mathbb{Z}, i \text{ odd}\}$$

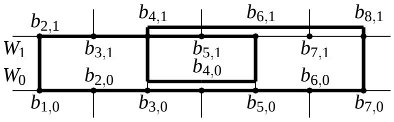


Fig. 6. A right-turn walk W in $\mathscr{B}$ with $t(W) = 2$.

Proposition 17 *Let W be a closed right-turn walk in $\mathscr{B}$. Then $t(W) \geq 1$. If $t(W) = 1$, then W has a clockwise solution.*

Proof: For a closed walk W in $\mathscr{B}$ of length k and any $l \in \{0, \ldots, k\}$ we define $t(W, l)$ to be the number of indices i with $1 \leq i \leq l$ such that W makes a right turn at i minus the number of those indices where W makes a left turn. So $t(W, 0) = 0$ and $t(W, k) = 6t(W)$, and for every l, $t(W, l+1) = t(W, l) \pm 1$.

Every closed walk in $\mathscr{B}$ contains a vertical edge (since walks do not turn back), so w.l.o.g. assume $W_0W_1 = b_{1,0}b_{2,1}$. Since W is a right-turn walk, after this a horizontal right edge follows, which is part of an alternating sequence of horizontal right and left edges. This sequence continues until a horizontal left edge is followed by a vertical edge, and after that an alternating sequence of horizontal right and left edges again follows (note that the walk cannot close before this point). So for some $i \geq 1$ and $j \geq 1$ we have

$$W_{0,2i+2j+2} = b_{1,0}, b_{2,1}, b_{3,1}, \ldots, b_{2i+2,1}, b_{2i+1,0}, b_{2i,0}, \ldots, b_{2i-2j+1,0}.$$

(In Figure 6, $i = 2$ and $j = 1$. Note that in general $j > i$ is also possible.) Choose j maximum, so either the walk W closes at this point ($k = 2i + 2j + 2$), or W continues with another vertical edge. In either case, the sequence $t(W,0), t(W,1), \ldots, t(W, 2i + 2j + 2)$ is then $0, (1,2)^i, 3, (4,5)^j, 6$. Continuing this reasoning shows that $t(W,l)$ can never decrease below 6 when $l \geq i + j + 2$, so we conclude $t(W) = t(W,k)/6 \geq 1$. In addition, if $t(W) = 1$, then the walk cannot contain another vertical edge, so

$$W = b_{1,0}, b_{2,1}, b_{3,1}, \ldots, b_{2i+2,1}, b_{2i+1,0}, b_{2i,0}, \ldots, b_{1,0}.$$

It is easily seen that in this case the subgraph of $\mathscr{B}$ induced by the vertices of W is a clockwise solution to W. □

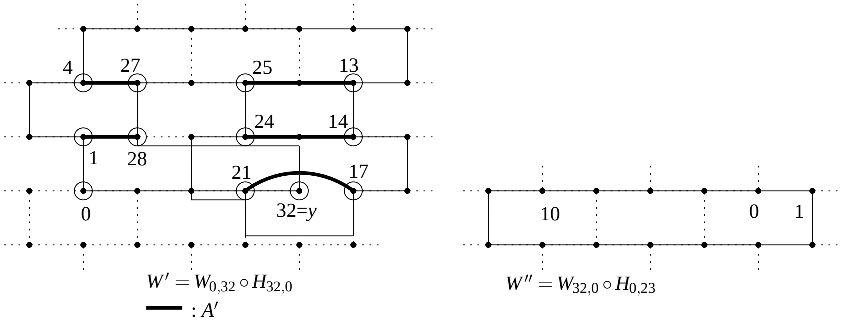

4
27
25
13
24
14
1
28
21
17
0
32=y
10
0
1
$W' = W_{0,32} \circ H_{32,0}$
: A'
$W'' = W_{32,0} \circ H_{0,23}$

Fig. 7. Splitting the walk W and assignment set A into two.

W'
W''

Splitting a walk In the following lemmas we use the following notations. Let A again be a proper assignment set for the closed walk W in $\mathscr{B}$. We assume A contains a PA $\{0,y\}$ such that $n(y) = 0$ (i.e. $W_{y,0}$ is a piece of W). We now consider the two new closed walks $W' = W_{0,y} \circ H_{y,0}$ and $W'' = W_{y,0} \circ H_{0,y}$ in $\mathscr{B}$. Note that W'' is in fact an elementary cycle of W, A, and that for all $i \in \{0, \ldots, y\}$, $W'_i = W_i$. Figure 7 illustrates this for the walk and assignment from Figure 5 (note that $y = 32$).

A'

We remark that although we consider various walks in $\mathscr{B}$ in this section, the notations $H_{i,j}$ and $n(i)$ are always defined with respect to W. Let $A' = A - \{0,y\}$. (We denote $A \backslash \{x\}$ and $A \cup \{x\}$ as $A - x$ and $A + x$ respectively.) Observe that i is a PA-index of W' if and only if it is a PA-index of W and $i \neq 0, y$. Thus it is easily seen that:

Proposition 18 *A' is a proper assignment set for W'.*

The above construction of smaller walks and corresponding assignment sets from a given pair W,A allows various induction proofs, justified by the next lemma.

Lemma 19 *Let W, A, W' and A' be as defined above. Then*

1. *For every elementary cycle C',I' of W',A', there is an elementary cycle C,I of W,A with $C \approx C'$.*
2. *W,A has one more elementary cycle than W',A'.*

Proof: We use the notation n' and $H'_{i,j}$ for the walk W', which are similar to the notations n and $H_{i,j}$ for W, so $n'(i)$ denotes the next PA-index of W' after i, and $H'_{i,j}$ denotes the horizontal path between W'_i and W'_j. Let C',I' be an elementary cycle of W',A' with

$$C' = W'_{i_0,n'(i_0)} \circ H'_{n'(i_0),i_1} \circ W'_{i_1,n'(i_1)} \circ \ldots \circ H'_{n'(i_l),i_0}.$$

All PAs in A' also appear in A, and all PA-indices of W' are PA-indices of W. It follows that C',I' is also an elementary cycle of W,A, unless one of the pieces, say $W'_{i_l,n'(i_l)}$ contains the part of W' corresponding to $H_{y,0}$. More precisely, this happens when $n(i_l) = y$ and $n'(i_l) = n(0)$. (In Figure 7, $n'(28) = 1 = n(0)$ but $n(28) = 32 = y$.) But in that case, we may replace $W'_{i_l,n'(i_l)}$ with $W_{i_l,y} \circ H_{y,0} \circ W_{0,n(0)}$, and choose the appropriate rotation (starting with 0), which yields an elementary cycle C,I of W,A with $C \approx C'$. This proves the first statement.

The above construction maps elementary cycles of W',A' to elementary cycles of W,A. It is easy to see that they are all mapped to different elementary cycles, which only contain pieces that are subwalks of $W_{0,y}$, and that every elementary cycle of W,A that contains a piece of $W_{0,y}$ is covered this way. It remains to consider elementary cycles of W,A that contain a piece of $W_{y,0}$. Since $\{0,y\} \in A$ and $n(y) = 0$, there is only one such elementary cycle C,I (with $C = W''$, as defined above). This proves the second statement. □

As a first application of Lemma 19, we can determine the number of elementary cycles.

Corollary 20 *Let A be a proper assignment set for closed walk W in $\mathscr{B}$, and let p be the number of PA-indices of W. Then the number of elementary cycles of W,A is $p/2+1$.*

Proof: If $A = \emptyset$ then there are no PA-indices, and W itself is the only elementary cycle, which proves the statement.

Otherwise we can use induction: choose a PA $\{x,y\} \in A$ with $x = n(y)$. Such a PA exists since A is a non-crossing perfect matching on the PA-indices. W.l.o.g we may assume that $x = 0$, since considering a different rotation of W and changing A accordingly does not change p or the number of elementary cycles.

Now consider W', A' and W'' as defined above using $\{0,y\}$. Let p' denote the number of PA-indices of W'. W'' is an elementary cycle of W, hence a right-turn walk, which has no PA-indices. Therefore every PA-index of W other than 0 or y corresponds to a PA-index in W', so $p'+2=p$. By Lemma 19 the number n of elementary cycles in W is equal to $n'+1$, where n' is the numbers of elementary cycles of W', A'. Since A' is a proper assignment set for W' (Proposition 18) we may use induction, so $n=n'+1=p'/2+2=p/2+1$. □

The above corollary will now be used to deduce that all elementary cycles defined by a *proper* assignment set have turning number 1, and hence admit a clockwise solution by Proposition 17.

Lemma 21 *Let A be a proper assignment set for a closed walk W in $\mathscr{B}$. Then $t(W)=1$ if and only if every elementary cycle C of A, W has $t(C)=1$*

Proof: The number of PA-indices of W is denoted by p. The number of left turns that W makes at non-PA-indices is denoted by $\textsc{Left}^*(W)$. Since W makes a left turn at every PA-index, we have

$$6t(W)=\textsc{Right}(W)-\textsc{Left}(W)=\textsc{Right}(W)-p-\textsc{Left}^*(W).$$

Let $\mathscr{C}$ denote the set of all elementary cycles of A, W. When summing the difference between right and left turns over all elementary cycles we obtain

$$\sum_{C\in\mathscr{C}}(\textsc{Right}(C)-\textsc{Left}(C))=\textsc{Right}(W)-\textsc{Left}^*(W)+2p.$$

Here we used the following observations. (i) Elementary cycles make right turns at PA-indices of W, and every PA-index contributes a right turn to two elementary cycles. (ii) Non-PA-indices of W contribute the same type of turn to one elementary cycle. (iii) All indices of elementary walks that do not correspond to pieces of W correspond to internal vertices of assignment paths; these vertices of assignment paths contribute a left turn to one elementary cycle and a right turn to another, hence these terms cancel. Combining this with $|\mathscr{C}|=p/2+1$ (Corollary 20) yields

$$6t(W)+3p=\textsc{Right}(W)-\textsc{Left}^*(W)+2p=$$

$$\sum_{C\in\mathscr{C}}(\textsc{Right}(C)-\textsc{Left}(C))=\sum_{C\in\mathscr{C}}(\textsc{Right}(C)-\textsc{Left}(C)-6)+3p+6\Longleftrightarrow$$

$$t(W)=\sum_{C\in\mathscr{C}}(t(C)-1)+1.$$

Because $t(C)-1\geq 0$ for all $C\in\mathscr{C}$ (Proposition 16, Proposition 17), this proves the statement. □

The proof of the next lemma is illustrated in Figure 8 (using the same example as before).

Lemma 22 *Let W, y, W' and W'' be as defined above. If S' and S'' are clockwise solutions of W' and W'' respectively, then a clockwise solution S of W exists with $\mathscr{A}(S)=\mathscr{A}(S')+\{0,y\}$*

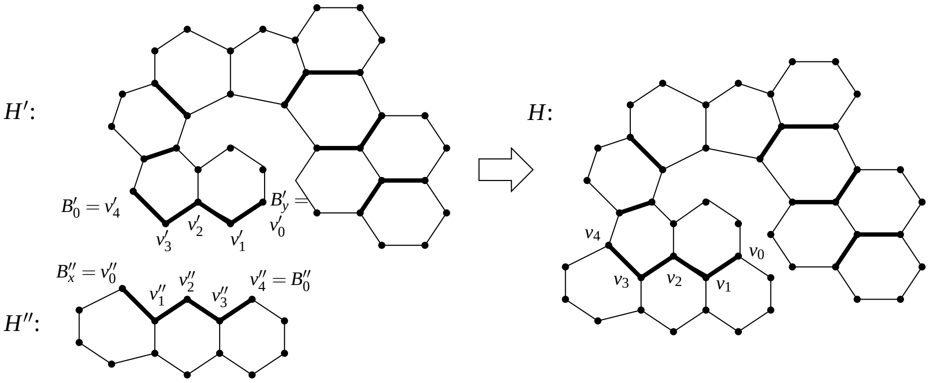


Fig. 8. Combining two partial solutions.

Proof: Let $S' = (H', \phi', B')$ be a clockwise solution to $W' = W_{0,y} \circ H_{y,0}$, and $S'' = (H'', \phi'', B'')$ be a clockwise solution to $W'' = W_{y,0} \circ H_{0,y}$. Let $x = k - y$ (recall that k is the length of W), so $W''_{0,x} = W_{y,0}$.

Let $B'_{y,0} = v'_0, v'_1, \ldots, v'_l$, and let $B''_{x,0} = v''_0, v''_1, \ldots, v''_l$. Note that these paths indeed have the same length l and

$$\phi'(v'_i) = \phi''(v''_{l-i}) \text{ for } i = 0, \ldots, l. \tag{1}$$

Since $\{0, y\}$ is a PA of W, we have $l \geq 1$. 0 and y are PA-indices, so W makes left turns at 0 and y, and therefore W' makes right turns at 0 and y, and W'' makes right turns at 0 and x. Because both S' and S'' are clockwise solutions, $d(v'_0) = d(v'_l) = 2$ and $d(v''_0) = d(v''_l) = 2$ follows. Since these two paths are mapped to the same paths in $\mathscr{B}$ but in reverse direction, and both S' and S'' are clockwise solutions, we see that v'_i has degree 2 when v''_{l-i} has degree 3 and vice versa, for $i = 1, \ldots, l-1$.

Construct H by starting with a copy of H' and a copy of H'', and for all $i \in \{0, \ldots, l\}$, identify the vertex v'_i of H' with the vertex v''_{l-i} of H''. Call the resulting vertex v_i. For $i \in \{0, \ldots, l-1\}$, this results in two parallel edges between v_i and v_{i+1}. Delete one edge of every such parallel pair. Because $l \geq 1$, this gives again a 2-connected graph. Using the above observations on the degrees of v'_i and v''_i, we see that $d(v_i) = 3$ for all i: either a vertex of degree 2 is identified with a vertex of degree 3 and two incident edges are removed (when $i = 1, \ldots, l-1$), or two vertices of degree 2 are identified and one incident edge is removed (when $i = 0, l$).

Choose an embedding of H in which every inner face of H' or H'' is again an inner face of H, and which has boundary cycle $v_l, B'_1, \ldots, B'_{y-1}, v_0, B''_1, \ldots, B''_{x-1}, v_l$. A LIH ϕ from H is constructed by setting $\phi(u) = \phi'(u)$ for all $u \in V(H) \cap V(H')$ and $\phi(u) = \phi''(u)$ for all $u \in V(H) \cap V(H'')$, and $\phi(v_i) = \phi(v'_i) = \phi(v''_{l-1})$ for the new vertices (see Equation (1)).

We observe that H is a hexagonal patch: all inner faces of H correspond to inner faces of H' or H'', and thus all have length 6. The degree constraints hold since the new vertices v_i have $d(v_i) = 3$ and boundary vertices of H' and H'' that were not identified remain boundary vertices.

Next we show that ϕ is a LIH. For vertices of H that have no neighbor in $V(H')\backslash\{v'_0,\dots,v'_l\}$ or no neighbor in $V(H'')\backslash\{v''_0,\dots,v''_l\}$, the local injectivity follows from the local injectivity of ϕ' and ϕ''. The only two vertices of H for which this does not hold are v_0 and v_l. But the three neighbors of v_l in H are mapped to W_1, W_{k-1} and a vertex on $H_{0,y}$, which are all different. A similar statement holds for v_0. This proves that ϕ is again a LIH.

We observed above that the boundary cycle of H is

$$B = v_l, B'_1, \dots, B'_{y-1}, v_0, B''_1, \dots, B''_{x-1}, v_l,$$

which is mapped by ϕ exactly to $W_0, W_1, \dots, W_{y-1}, W_y, W_{y+1}, \dots, W_{k-1}, W_0 = W$. So (H,ϕ,B) is a solution to W. Since we started with clockwise solutions, it is also obviously a clockwise solution. (By Lemma 1 we do not have to check this for all boundary vertices.)

An assignment path from B'_i to B'_j in H' yields an assignment path from B_i to B_j in H. In addition, the path $v_0,\dots,v_l$ in H is mapped by ϕ to $H_{y,0}$, and no internal vertices of this path lie on the boundary of H. Hence this solution assigns 0 to y. Now all PA-indices of W are accounted for (W'' contains no PA-indices), which shows that $\mathscr{A}(S) = \mathscr{A}(S') + \{0,y\}$. □

Finally, we have collected all the ingredients that are necessary to prove the remaining lemma.

Proof of Lemma 9: We show that if a proper assignment set A is given for a walk W with $t(W) = 1$, then a clockwise solution S of W with $\mathscr{A}(S) = A$ exists.

If $A = \emptyset$ then since A is a proper assignment set, this implies that W has no PA-indices and W itself is the only elementary cycle of A,W. This is a right-turn walk by Proposition 16, and since $t(W) = 1$, it has a clockwise solution S by Proposition 17. Note that $\mathscr{A}(S) = \emptyset = A$ since there are no PA-indices.

If $A \neq \emptyset$, then choose an arbitrary $\{x,y\} \in A$ with $n(y) = x$ (this exists since A is a non-crossing perfect matching). We may again assume w.l.o.g. that $x = 0$. Use this to define W', W'' and A' as before. A' is a proper assignment set for W' (Proposition 18). Lemma 19 shows that for every elementary cycle C' of W',A', an elementary cycle C of W,A with $C \approx C'$ exists. By Lemma 21, $t(C) = 1$ follows from $t(W) = 1$, so $t(C') = 1$ for every such elementary cycle C'. Therefore by applying Lemma 21 again, we obtain $t(W') = 1$. At this point we have a new closed walk W' in $\mathscr{B}$ with turning number 1, with fewer PA-indices, and a proper assignment set A' for it. Hence by induction, W' admits a clockwise solution S' with $\mathscr{A}(S') = A'$. W'' itself is an elementary cycle of W, so $t(W'') = 1$ (Lemma 21), and therefore W'' admits a clockwise solution S'' as well (Proposition 16, Proposition 17). Now Lemma 22 shows that S' and S'' can be combined into a clockwise solution S for W with $\mathscr{A}(S) = A$. □

C Proofs of Section 5

Proof of Lemma 13: We show that subroutine $S(i,j)$ correctly calculates $S_{i,j}$ when the stated $S_{x,y}$ values are known, in time $O(d(i))$. Clearly, the algorithm only uses values $S_{x,y}$ with $-1 \leq y - x < j - i$ for the calculations.

Observe that throughout the algorithm, the value of m equals the size of some PM of $G_{i,j}$. We now show that in some line, the size of a MPM is considered, which proves correctness.

Let M be a MPM of $G_{i,j}$. If $j \leq i$ then $M = \emptyset$ which is considered in line 1. Now suppose $j > i$. If M contains no edge incident with i, it is a PM of $G_{i+1,j}$ and considered in line 2. Otherwise, let $iv \in M$. If $v = j$, then M consists of this edge and a PM of $G_{i+1,j-1}$, which is considered in line 6. Otherwise, because M is non-crossing, it can be partitioned into a MPM of $G_{i,v}$ and $G_{v+1,j}$, which is considered in the for-loop.

The complexity of the algorithm is determined by the for-loop, which iterates at most $d(i)$ times. □

Proof of Lemma 14: We show that subroutine $N(i, j)$ correctly calculates $N_{i,j}$ when the stated $N_{x,y}$ and $S_{x,y}$ values are known, in time $O(d(i))$. The algorithm only uses values $S_{x,y}$ with $-1 \leq y - x \leq j - i$ and values $N_{x,y}$ with $-1 \leq y - x < j - i$ for the calculations.

Clearly, line 1 returns the correct answer (in this case the empty set is the unique MPM). Otherwise, the algorithm adds x to the number N whenever x different MPMs of $G_{i,j}$ are found that have not been considered earlier. We show that all cases are considered and no MPMs are double counted, which shows that the correct answer is returned in line 7.

If $S_{i,j} = S_{i+1,j}$ then all MPMs of $G_{i+1,j}$ are also MPMs of $G_{i,j}$, which explains line 3. This accounts for all MPMs of $G_{i,j}$ that do not contain an edge incident with i. $G_{i,j}$ admits MPMs M that contain an edge ij if and only if $S_{i,j} = w_{ij} + S_{i+1,j-1}$. Then $M - ij$ is a MPM of $G_{i+1,j-1}$, and there is a bijection between such MPMs, which explains line 6. It remains to consider MPMs M of $G_{i,j}$ which contain an edge iv with $i+1 \leq v \leq j-1$. These can be decomposed into a PM of $G_{i,v}$ and a PM of $G_{v+1,j}$, which must be MPMs, so $S_{i,j} = S_{i,v} + S_{v+1,j}$. In fact, if $S_{i,j} = S_{i,v} + S_{v+1,j}$ then every combination of a MPM of $G_{i,v}$ and a MPM of $G_{v+1,j}$ gives a unique MPM of $G_{i,j}$ that contains iv, which explains line 5. All cases are now considered, so at the end of the algorithm, $N = N_{i,j}$.

The complexity of the algorithm is determined by the for-loop, which iterates at most $d(i)$ times. □